**Cybersecurity in African E-Government Services: A Vulnerability Assessment and Recommendations Report**

### 

**Technical Collaborators:**

Byagutangaza Theoneste [tbyaguta@andrew.cmu.edu](mailto:tbyaguta@andrew.cmu.edu)

Emmanuel Agbragu [eagbragu@andrew.cmu.edu](mailto:eagbragu@andrew.cmu.edu)

Junias Bonou [jbonou@andrew.cmu.edu](mailto:jbonou@andrew.cmu.edu)

Lenah Chacha [lchacha@andrew.cmu.edu](mailto:lchacha@andrew.cmu.edu)

Assane Gueye [assaneg@andrew.cmu.edu](mailto:assaneg@andrew.cmu.edu)

**Date: 01/03/2025**

### 

# Table of Contents

[**Table of Contents 2**](#_d7ylymjx6i8)

[**Glossary of Terms 3**](#_xburamzhkyx1)

[**About the Report 3**](#_h865lk3p5281)

[**Executive Summary 4**](#_nx448f6m3scb)

[**Introduction 7**](#_38zm2evc7nh)

[**Scope And Methodology 8**](#_7qkcx3be0es6)

[Enumeration of government e-services: 8](#_8idbdzdiuf67)

[Vulnerability Assessment and Penetration Testing: 8](#_z5eiwvv4jyax)

[Assessment Frameworks 8](#_6dqdza2x5m50)

[Risk Prioritization Criteria 9](#_zd9pbl4haphf)

[Advantages and Disadvantages of our methodology 9](#_xpoa4d1vdfi1)

[Tools Utilised 9](#_5m1xu1fahbc2)

[**Result Evaluation 11**](#_ndsnq4k6dq3w)

[Vulnerabilities Summary 12](#_fhohcjg9h9xj)

[Total Urls Tested Per region 13](#_qtr68zu13rrs)

[**Recommendations and Best Practices 14**](#_7ujd0zn9k9qt)

[**Stakeholder Engagement 15**](#_kq5dfplhw49i)

[**Conclusion 15**](#_2k73lecz2edk)

[**References 15**](#_uq32ohqk2kuw)

[**Appendices 16**](#_ceu86rnkybuk)

### 

# 

# Glossary of Terms

| **Abbreviation** |  |
| --- | --- |
| OWASP | Open Web Application Security Project |
| NIST | National Institute of Standards and Technology |
| CWE | Common Weakness Enumeration |
| XSS | Cross-Site Scripting |
| HTTP | Hypertext Transfer Protocol |
| SQL | Structured Query Language |
| CISA | Cybersecurity and Infrastructure Security Agency |
| XST | Cross-Site Tracing |

# 

# Acknowledgment

We would like to extend our heartfelt gratitude to our cyber security experts and analysts in our lab for their constructive feedback, their technical contribution, and their guidance throughout the project that made it possible to have this assessment. This collaborative effort has strengthened the importance of proactive security assessment to the government's digital services, and we look forward to continuing to secure the government e-services.

# 

# 

# 

# 

# 

# 

# 

# 

# About the Report

This report is intended for government officials, policymakers, and cybersecurity professionals involved in the digital transformation and security of e-government services across Africa. It aims to provide them with insights into the current cybersecurity posture of government-owned e-services, highlight key vulnerabilities, and offer actionable recommendations to enhance security and resilience. The report can also be useful for organizations and stakeholders involved in supporting and advising on cybersecurity measures for public administration.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

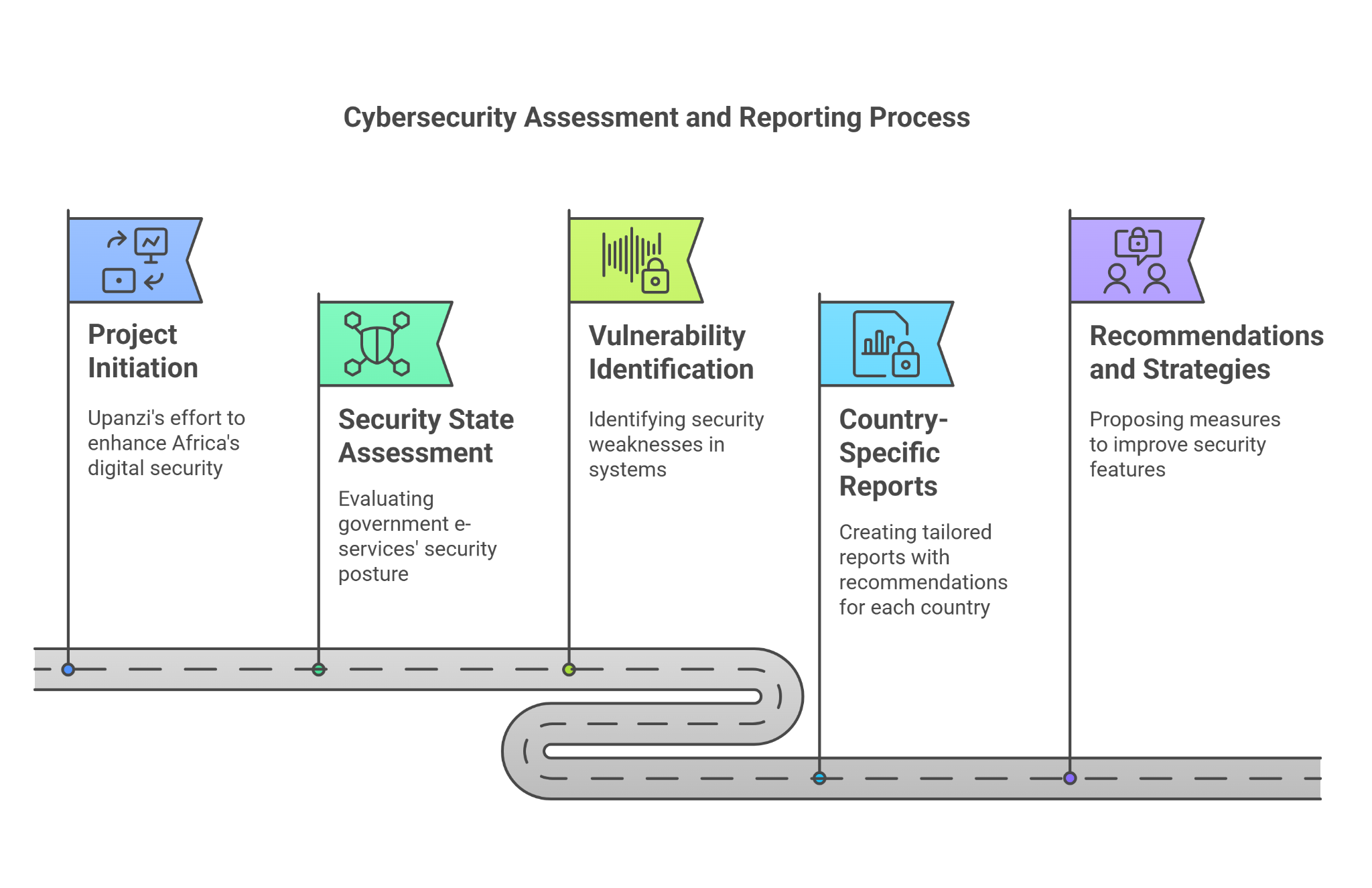
# Executive Summary

This project is part of Upanzi’s effort to contribute to building a secure and resilient digital transformation of Africa. More precisely, it aims to understand the overall state of security of government-owned e-services in Africa and their cybersecurity posture and preparedness to handle security threats. Measuring our current cybersecurity posture is a major step in compiling the necessary mitigation measures to lead to an enhanced security environment. For over four months, we have conducted several tests of minimal threat over publicly facing government e-services across Africa. In our endeavor, the current status of security was reviewed, and vulnerabilities were identified. A separate report was then written for each country, which included recommendations and strategies to establish and improve the security features of the different digital services offered to the citizens.

Recognizing the generally strong security posture observed in most of the assets evaluated in African cyberspace is essential. Most of the infrastructure demonstrated a commendable level of robustness, with adequate security controls and practices that mitigated several common risks. This reflects the proactive approach to safeguarding critical services and protecting sensitive data. These solid foundations provide an excellent starting point for addressing the identified vulnerabilities, further strengthening the overall security of the environment.

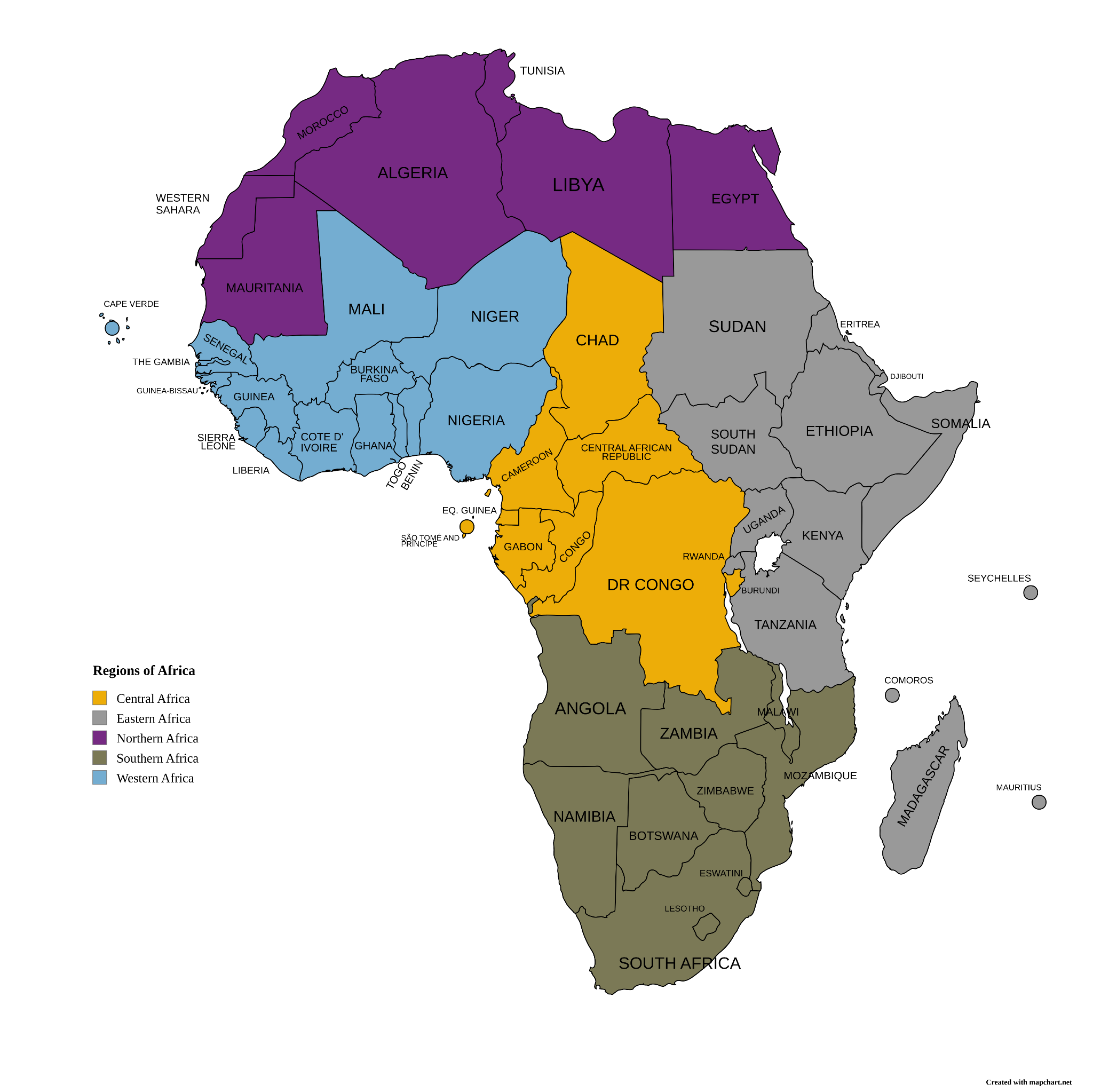
The vulnerability assessment of the government’s African websites surveyed over **77,000** assets (unique domains). It revealed important security issues, including Weak credentials, Cross-Site Scripting (XSS), insecure HTTP methods being enabled, and some instances of sensitive content exposure. These vulnerabilities pose significant risks, potentially leading to unauthorized access, data breaches, and other security threats, necessitating immediate remediation to protect the integrity and confidentiality of the government's digital services.

Below is a summary of the vulnerabilities discovered on target hosts ( discoverable government domains).



*Figure 1: Infographic showing the roadmap of the project*

| Vulnerability | OWASP top 10 | Severity |
| --- | --- | --- |
| Sensitive Data Exposure - archive file | Sensitive data exposure | Critical |
| Sensitive Data Exposure - Database backup file | Sensitive data exposure | Critical |
| Sensitive Data Exposure - .git folder | Sensitive data exposure | Critical |
| Weak WordPress credentials | Broken authentication | Critical |
| SQL Injection | Injection | Critical |
| Cross-site scripting (XSS) - Reflected | Injection | High |
| Sensitive Data Exposure - Log files | Sensitive data exposure | Medium |
| TRACE Method Enabled | Security Misconfiguration | Medium |

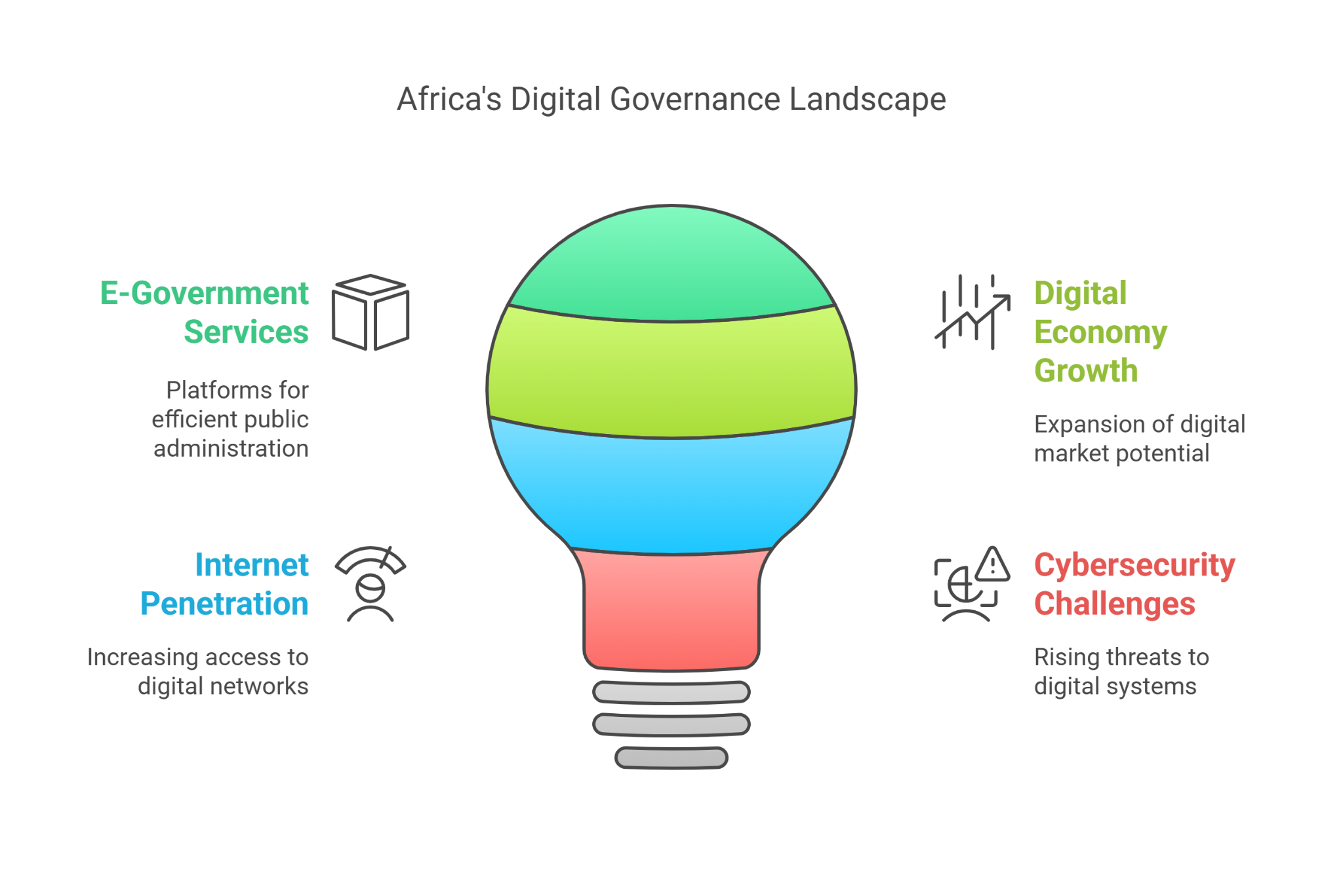
The test covered all African countries grouped into five distinct regions according to the African Union:

***Fig1:*** *AU Member States Regional Groups*

*(*[*https://au.int/sites/default/files/documents/31829-doc-AU\_HBK\_2021\_-\_ENGLISH\_web.pdf*](https://au.int/sites/default/files/documents/31829-doc-AU_HBK_2021_-_ENGLISH_web.pdf) *)*

# 

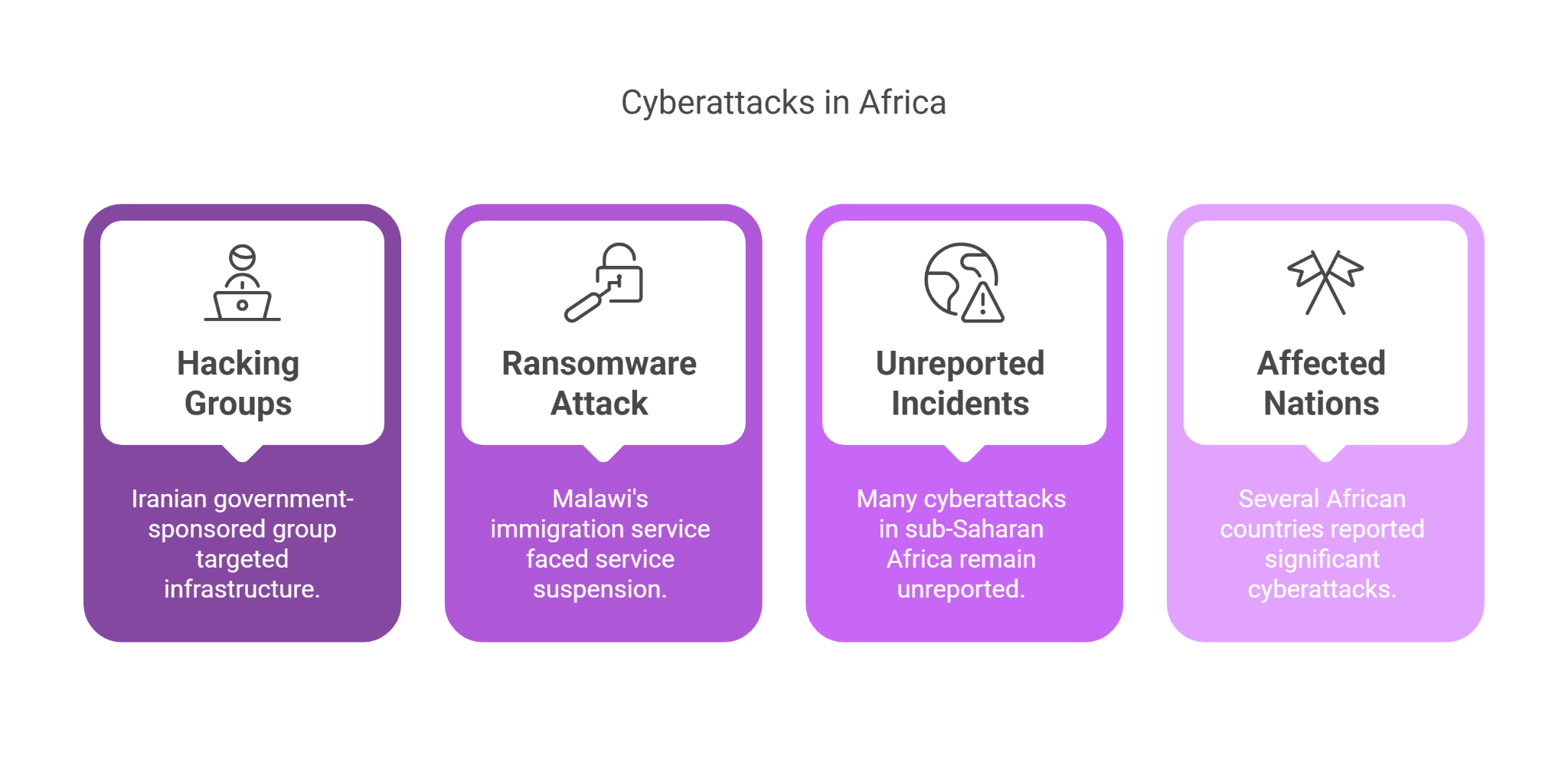
# Introduction

Across Africa, governments are rapidly adopting digitalization to deliver services more efficiently through e-government platforms. From tax recording and business and land registrations to identity management, these systems are becoming the focal point of public administration. E-government services adoption in Africa is predicted to grow because of the continent's expanding digital economy and increasing internet penetration. A 2020 study by Google and the International Finance Corporation (IFC) projected that Africa's digital economy is expected to reach US$712 billion (8.5% of GDP) by 2050[5]. However, this shift towards digital governance has introduced new and significant security challenges. Consequently, cybercrime was expected to cost over USD 4 billion in 2021, accounting for about 10% of the continent's GDP. The threat has grown in volume, impact, and complexity, with nearly half of the surveyed African countries experiencing ransomware attacks on critical infrastructure—such as government systems, hospitals, and financial institutions—between January and December 2023[6].

*Figure 2: Showing the trend of the digital landscape in Africa*

In recent years, cyberattacks targeting e-government platforms have been on the rise. These attacks have led to the exposure of personal information, identity theft, and financial fraud. In 2023, *eCitizen*—Kenya's citizen-centric information and services platform and the country’s official government digital payment portal—was rendered inaccessible due to a Distributed Denial-of-Service (DDoS) attack. This disruption halted critical services, including visa issuance [1]. At the same time, other vital systems, such as the national electricity transmission and distribution provider, Kenya Power, and Kenya Railways' passenger services, also suffered outages due to similar DDoS attacks. As a result, citizens could not purchase railway tickets or buy electricity units during the downtime [1].

In the same year, *CyberAv3ngers*, an Iranian government-sponsored hacking group, was allegedly responsible for attacks on South Africa’s water and sewage control systems. This incident followed a warning by the U.S. Cybersecurity and Infrastructure Security Agency (CISA), which reported that the group had exploited security weaknesses in industrial control systems. The attackers are believed to have gained access to Unitronics Vision Series Programmable Logic Controllers (PLCs) with Human Machine Interfaces (HMIs), by taking advantage of poor password practices and other security vulnerabilities [2]. In 2024, Malawi’s immigration service was hit by a ransomware attack, resulting in a three-week suspension of passport issuance services [3]. Other African nations—including Uganda, Rwanda, Ghana, Nigeria, and Zimbabwe—reported significant cyberattacks in 2024. Additionally, GIS Reports note that many cyberattacks in sub-Saharan Africa go unreported, suggesting that the continent's vulnerability is more significant than current estimates indicate[4].



*Figure 4: Showing effects of cybersecurity attacks on nations*

To help African governments improve the cybersecurity posture of their e-services, the Upanzi Network team conducted vulnerability assessments and penetration testing. The goal was to assess the security of government e-services and provide recommendations to fix identified vulnerabilities. Testing covered all African countries, divided into five regions according to the African Union: Central, Eastern, Northern, Southern, and Western Africa. Services were tested against standard frameworks like OWASP and CWE, and results and recommendations were shared with some countries to address security flaws.

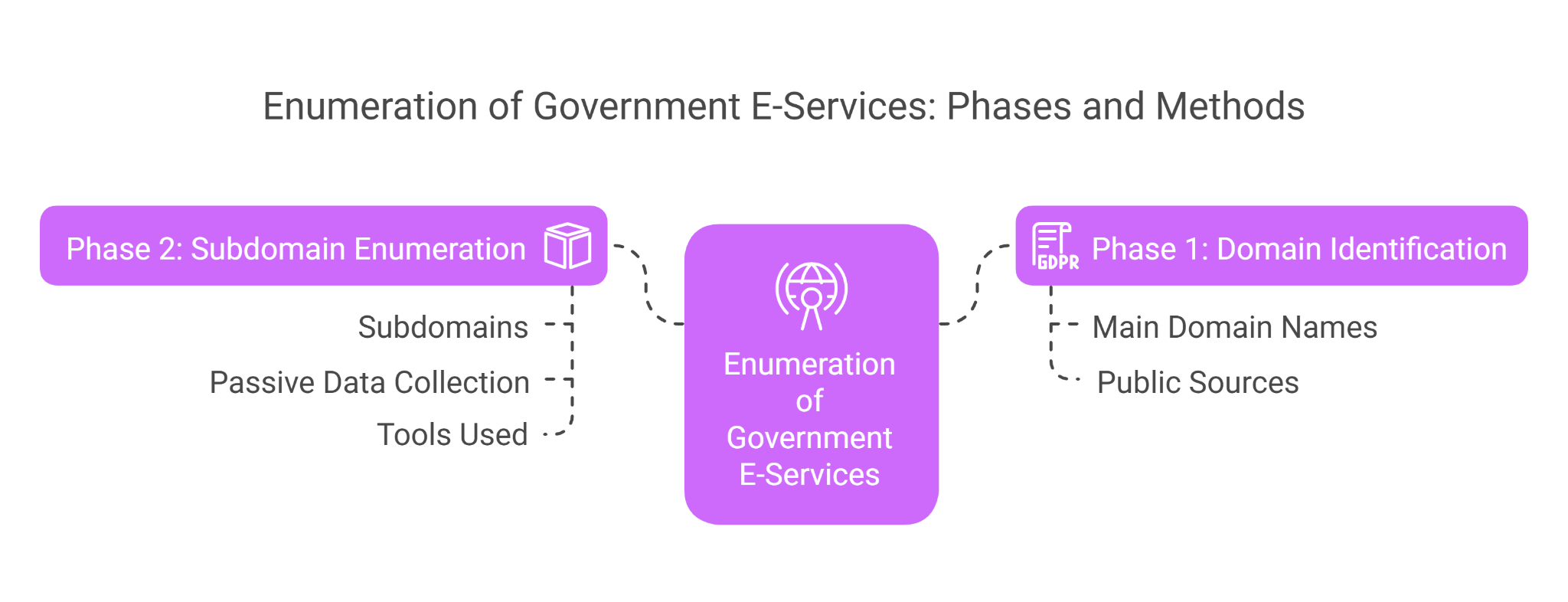
# 

# Scope And Methodology

The project is implemented in 2 phases.

## Enumeration of government e-services:

A round of enumeration of government-owned e-services across the continent took place. It consisted in first obtaining the main government domain name (e.g., gov.rw, go.ke, gouv.ga, etc.) and getting the list of government-owned e-services from enumerating respective subdomains of those. This enumeration is fully passive and requires no interaction with the targets. The data is fetched from public sources (google, Netcraft, virustotal, crt.sh, PassiveDNS, etc.).

*Figure 5: Diagram showing the phase methods used under this assessment* 

## Vulnerability Assessment and Penetration Testing:

After listing government e-services across the continent, a set of automated tests were conducted. It is important to stress the fact that the study required a security assessment plan which defined:

* An assessment checklist coherent with the goals and objectives while posing a minimal threat to the target e-services' availability and continuity: at this stage, an exhaustive list of common vulnerabilities from different frameworks (OWASP, CWEs, etc.).



### 

*Figure 6: showing the vulnerability assessment and penetration testing plan*

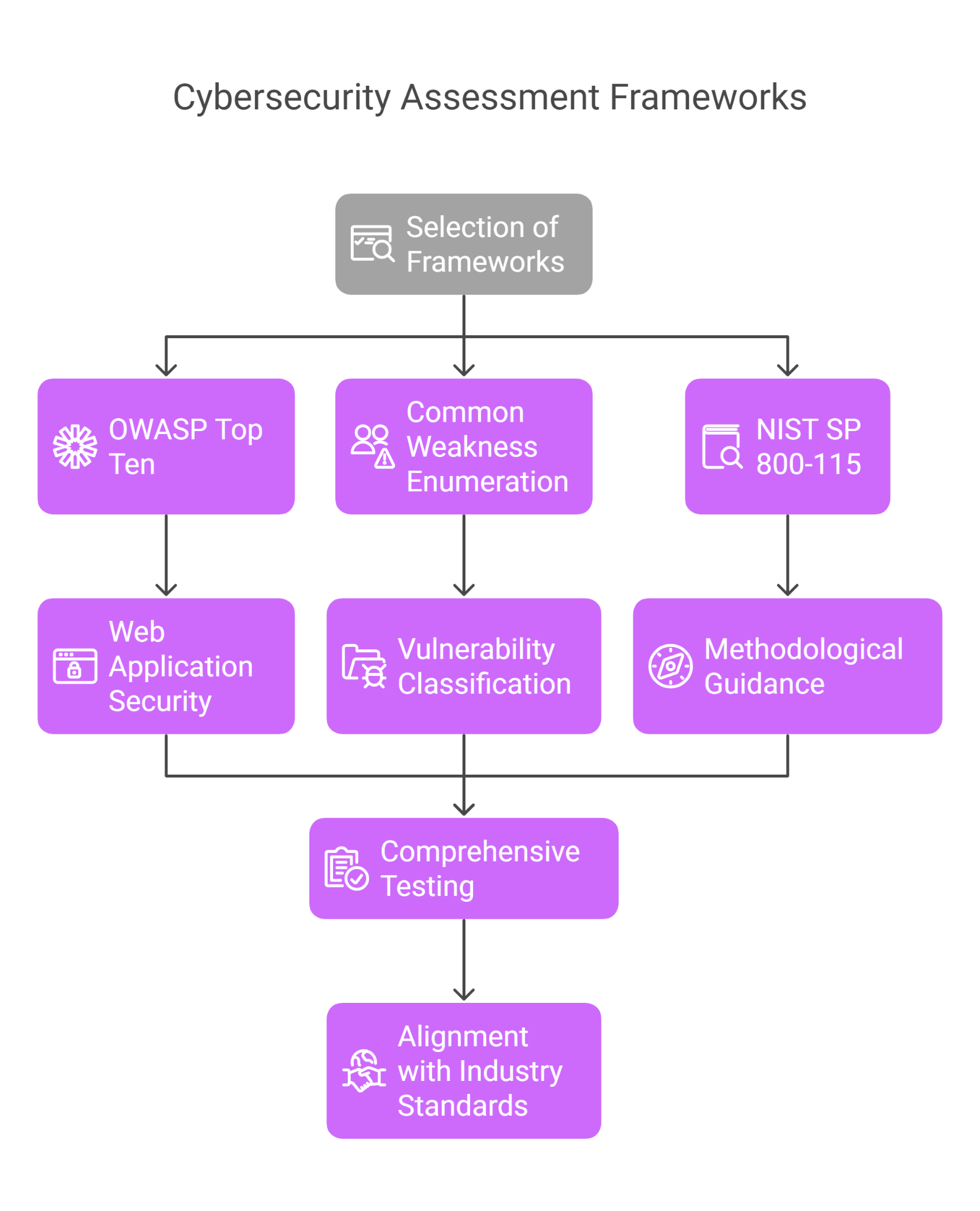
### 

### Assessment Frameworks

The assessment was conducted based on these widely recognized security frameworks, including:

* **OWASP Top Ten (2021):** It is an industry standard for web application security, relevant to common web application vulnerabilities, and provides more suitable baselines for testing public services.
* **Common Weakness Enumeration (CWE):** Used to provide a more efficient vulnerability classification for software.
* **NIST SP 800-115 (Technical Guide to Information Security Testing and Assessment):** Provided methodological guidance to ensure the assessments followed industry best practices for non-intrusive testing and accurate reporting.

These frameworks were chosen due to their wide acceptance, comprehensive guidance, and alignment of industry standards for testing and risk management of cyber security services, including web applications.



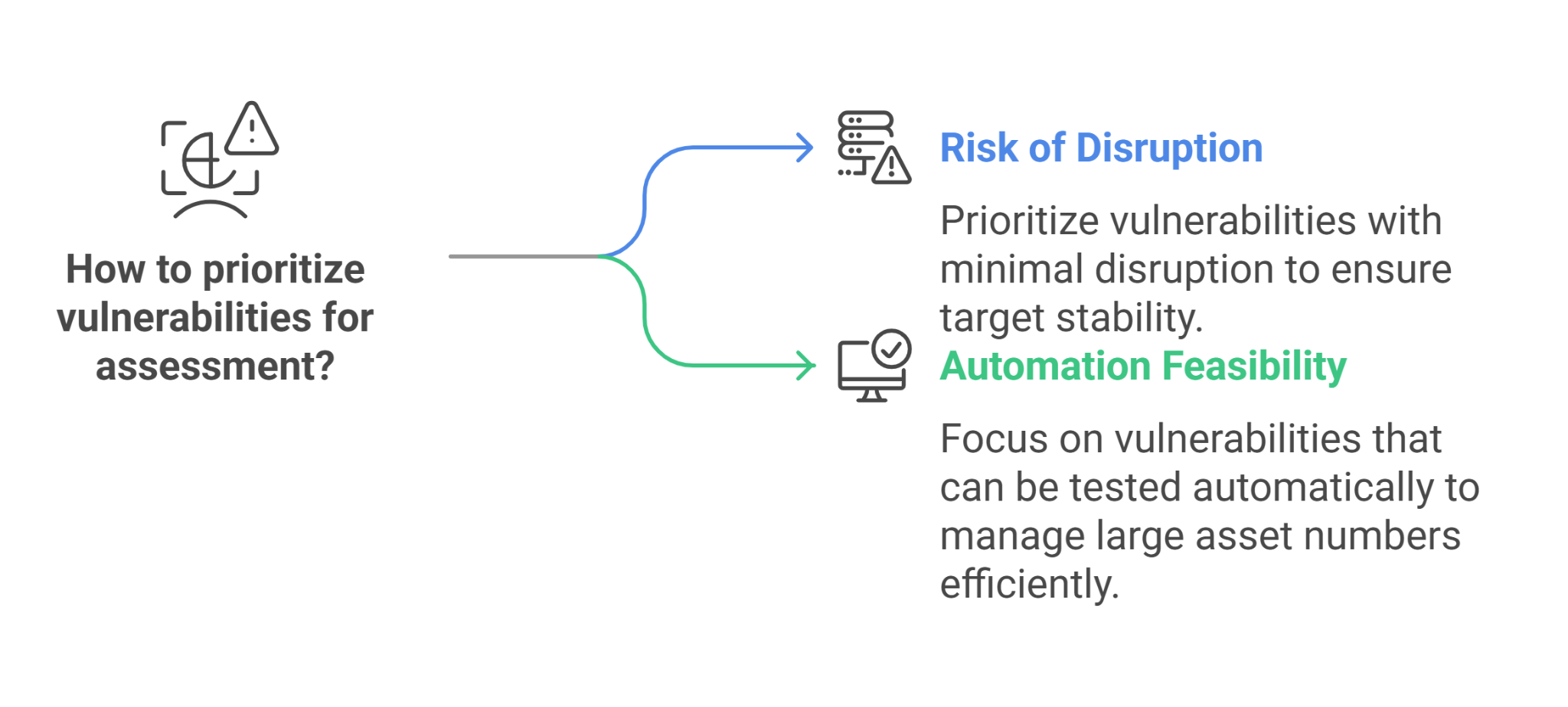
*Figure 7: Diagram showing the cybersecurity assessment frameworks used for this project.*

### Risk Prioritization Criteria

A formal validation was followed to prioritize which vulnerabilities to assess based on two primary criteria:

**Risk of Disruption:** Vulnerabilities that cannot be tested without risking noticeable damage or disruption to the target are discarded, and priority is given to those with the fewest disruptions.

**Automation Feasibility:** Due to the large number of assets, only vulnerabilities that can be tested automatically using scripts or tools are kept. All others were discarded. Strategies for testing, result collection, and result evaluation that are the least intrusive possible for the target.



*Figure 8: Diagram depicting the risk prioritization criteria*

#### Advantages and Disadvantages of our methodology

**Advantages:**

One key advantage of this methodology is the ability to assess the vulnerabilities of these assets across multiple regions while minimizing risks.

* Scalable to a large number of assets across different countries.
* Disruption avoidance: minimizes risks and focuses on passive enumeration and non-disruptive testing.
* Followed widely recognised security frameworks making findings and structure more credible.

**Disadvantages**

The disadvantages of this methodology include its limitations in addressing complex types of vulnerabilities and possible underestimation of some risks.

* Our testing is not suitable for complex vulnerabilities like business logic flaws and chained vulnerabilities because they require active testing.
* Avoidance of active vulnerabilities enumeration might result to vulnerabilities scoring underestimates certain risks.

## 

## Tools Utilised

The automated tests were carried out using tools and scripts written by the team to have complete control and visibility over all the interactions with the target as part of the tests. Some of the scripts invoke preexisting modules and tools described below.

| Tools | Description | Selection criteria |
| --- | --- | --- |
| archivecheck.py | A Python script to look for backups and archives of the website that could have been left in the web root | It identifies the unintended exposure of sensitive files containing personal information and system configurations.  **Relevance**: These files are often overlooked during cleanup, making them a target for hackers. Using scripting helps to tailor the search using regex to reduce false positives that is often prevalent in some open source tools. |
| content-discovery.sh | A shell script designed to discover hidden files and directories on a web server by probing common file names and extensions. | **Relevance**: Automated tools might miss the custom path and they are very slow as they contain more configurations that are not relevant to our search, but with this script, we customized it with the organization’s settings. |
| git\_checker.py | Python script to look for Git repositories and git objects left accessible in the web root. | This script detects the publicly available git repository, this file might reveal the source codes and other confidential data.  **Relevance**: Exposing git repository can pose significant security breaches. Scripting can make it possible to include organizations specific development culture. |
| http\_methods\_assessment.py | A script that tests which HTTP methods (e.g., GET, POST, PUT, DELETE) are enabled and accessible on a web server. | This script helps to identify the HTTP methods that can be exposed when is misconfigured.  **Relevance**: Ensuring only necessary HTTP methods are enabled to reduce the breaches that might arises. |
| sqli-checker.py | A script that tests web applications for potential SQL injection vulnerabilities by including special characters in GET parameters. | **Relevant**: SQL injection is prevalent today and it remains a critical vulnerability. The tools available are very slow because they have many parameters that we didn’t need, this script was tailored to our testing contexts. |
| traceCheck.py | A script that checks for HTTP TRACE method vulnerability, which can lead to cross-site tracing attacks or session hijacking in web applications. | **Relevant**: Disabling unnecessary HTTP methods like trace methods are recommendable practices, we had to develop this script as there is no such tool specifically for trace methods detection. |
| wp-creds.sh | A script targeting WordPress installations, possibly attempting to discover admin credentials by checking for common username-password combinations. | Many e-gov services we tested were built in WordPress. The available tools like wpscan are not efficient for checking vulnerabilities on many websites at the same time due to the slowness. |
| xss.py | A Python script that tests for Cross-Site Scripting (XSS) vulnerabilities in web applications, aiming to identify improperly sanitized GET parameters that could allow for XSS exploits | Cross-site scripting(XSS) it is prevalent today and it remains a critical vulnerability. The tools available are very slow because they have many unnecessary parameters. |

**RISK ASSESSMENT MATRIX**

A risk assessment matrix is a tool used in risk management to evaluate and prioritize risks. It helps to present which risks need more attention and which are less critical.

We used the OWASP Risk Rating framework and the OWASP Risk Rating Calculator for this risk assessment matrix. This framework provides guidelines for risk assessments, often using a number scoring system to prioritize risks.

Within this matrix, we define four key components:

* **Vulnerability**: List of the security weaknesses identified during our assessment.
* **Likelihood (L)**: Represents the probability of a vulnerability being exploited.
* **Impact (I)**: Measures the potential damage if the vulnerability is successfully exploited.
* **Risk Level (L × I)**: Represents the overall risk posed by a vulnerability, calculated as the product of Likelihood and Impact.

By creating a risk assessment matrix, we systematically identified vulnerabilities in the web services and assessed their likelihood of exploitation. This was done by categorizing the likelihood as "Rare," "Possible," or "Likely" based on factors such as the skill level, motive, opportunity, and size of potential attackers, as well as the ease of discovery and exploitation of the vulnerabilities. For instance, SQL injection is highly likely to be exploited due to its technical simplicity, the attacker’s potential motives (such as stealing or selling data), and the ease of access, with no authentication required. In evaluating the overall risk, both the technical and business impacts must be considered. The technical impacts include the potential loss of critical data or system corruption, while the business impacts can be financial loss and reputational damage. By understanding these factors, we can determine the severity of the risk and make informed decisions about necessary mitigation measures.



# 

# Result Evaluation

The team manually scrutinized and tested all the results to confirm the vulnerabilities' existence and exploitability.

## Vulnerabilities Summary

| **Vulnerability** | **Description** |
| --- | --- |
| **Injection** | In our assessment, we identified injection vulnerabilities such as SQL injection and reflected-XSS vulnerabilities. The risk associated with injection vulnerabilities depends on the vulnerable code and its role within the application. For example, SQL injection vulnerabilities are commonly exploited to access sensitive data stored within an SQL database[7]. In our assessment, we found websites vulnerable to SQL injection, and its exploitation could result in potential harm to the data, such as data breaches or unauthorized data deletion and modification. Additionally, Cross-Site Scripting (XSS) vulnerabilities were found in some of the assessed web applications due to improper sanitization of user-supplied input in the GET parameter. This vulnerability allows an attacker to inject malicious scripts into the website, which can then be executed in the context of the user's browser. |
| **Sensitive Data/Files Exposure** | Sensitive files are publicly accessible or improperly protected, leading to data leaks. These range from citizen's PIIs to cryptographic keys and passwords. These sensitive files are .git files, archive files, log files, and database backup files. These files contain highly sensitive information that can compromise web applications, associated databases, and users and government agencies that maintain them, and they were found to be publicly accessible.  The most sensitive files uncovered in those archives include:   * websites’ source code * database passwords * database backups * WIFI passwords * Users and administrator's passwords (cleartext and hashes) * Scans of National ID * Other personally identifiable information(Home addresses, Email addresses, CV, diplomas, photographs, ) and other documents   If an attacker gains access to any of these exposed files, they can obtain full knowledge of the web application’s source code, configuration, and credentials, leading to a complete compromise of the system. This can result in unauthorized access to databases, user data, and the ability to modify or deface the site. They may as well leverage the personally identifiable information exposed for impersonation and identity theft purposes. |
| **Broken Authentication** | A couple of the applications we tested were using WordPress technology and were prone to broken authentication attacks due to the use of weak passwords that are easily predictable. Weak passwords, such as short passwords, common words, or simplistic patterns, can be cracked using brute force or dictionary attacks. These attacks involve automated tools that systematically guess passwords until the correct one is found. In WordPress, weak passwords pose a high risk, especially for users with administrative privileges, as gaining access to these accounts allows attackers to fully compromise the website, including its content, plugins, themes, and settings. If an attacker successfully exploits this vulnerability, they can gain unauthorized access to the WordPress dashboard, which may lead to a full site takeover. The attacker could modify website content, inject malicious scripts (e.g., malware or phishing code), delete critical data, or install backdoors to maintain persistent access. A compromised admin account could also lead to data breaches, exposing sensitive user information. Additionally, the website’s reputation may suffer, and it could be blacklisted by search engines or flagged by security services, resulting in traffic and revenue loss. |
| **Security Misconfiguration( TRACE HTTP Methods enabled)** | The HTTP TRACE method is enabled on different sites. TRACE is a diagnostic method that echoes the received request, which can be exploited in Cross-Site Tracing (XST) attacks. This can be used in conjunction with XSS to steal cookies and authentication tokens from legitimate users. Enabling the TRACE method increases the risk of an XST attack, potentially leading to unauthorized access to sensitive information such as session cookies. The attacker could use this information to impersonate legitimate users, including administrators. |

These vulnerabilities pose significant risks, potentially leading to unauthorized access, data breaches, and other security threats, necessitating immediate remediation to protect the integrity and confidentiality of the government's digital services.



*Figure 9: Diagram depicting the overall ranking of vulnerabilities identified in this research project.*

## Total Urls Tested Per region

This graph presents the number of systems tested per region, based on the African Union's regional divisions. Eastern and Western Africa each consist of 15 countries, while Southern Africa has 10, Central Africa has 9, and Northern Africa has 7. The higher number of countries in Eastern and Western Africa likely contributed to the detection of more e-government services compared to other regions. Notably, Central Africa has the fewest e-government systems identified. This might be attributed not only to the region's smaller number of countries but also to its lower E-Government Development Index (EGDI 2024) score[8], which suggests limited digital infrastructure and fewer online public services.

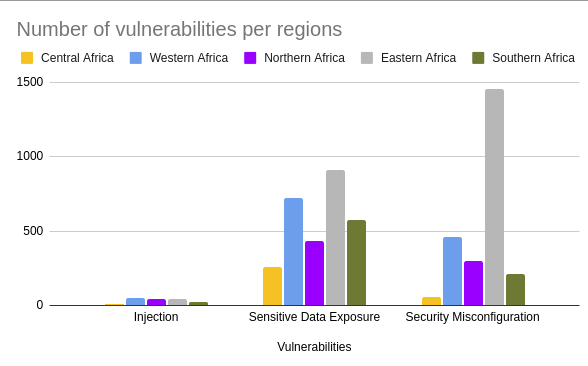
****

**Vulnerabilities**

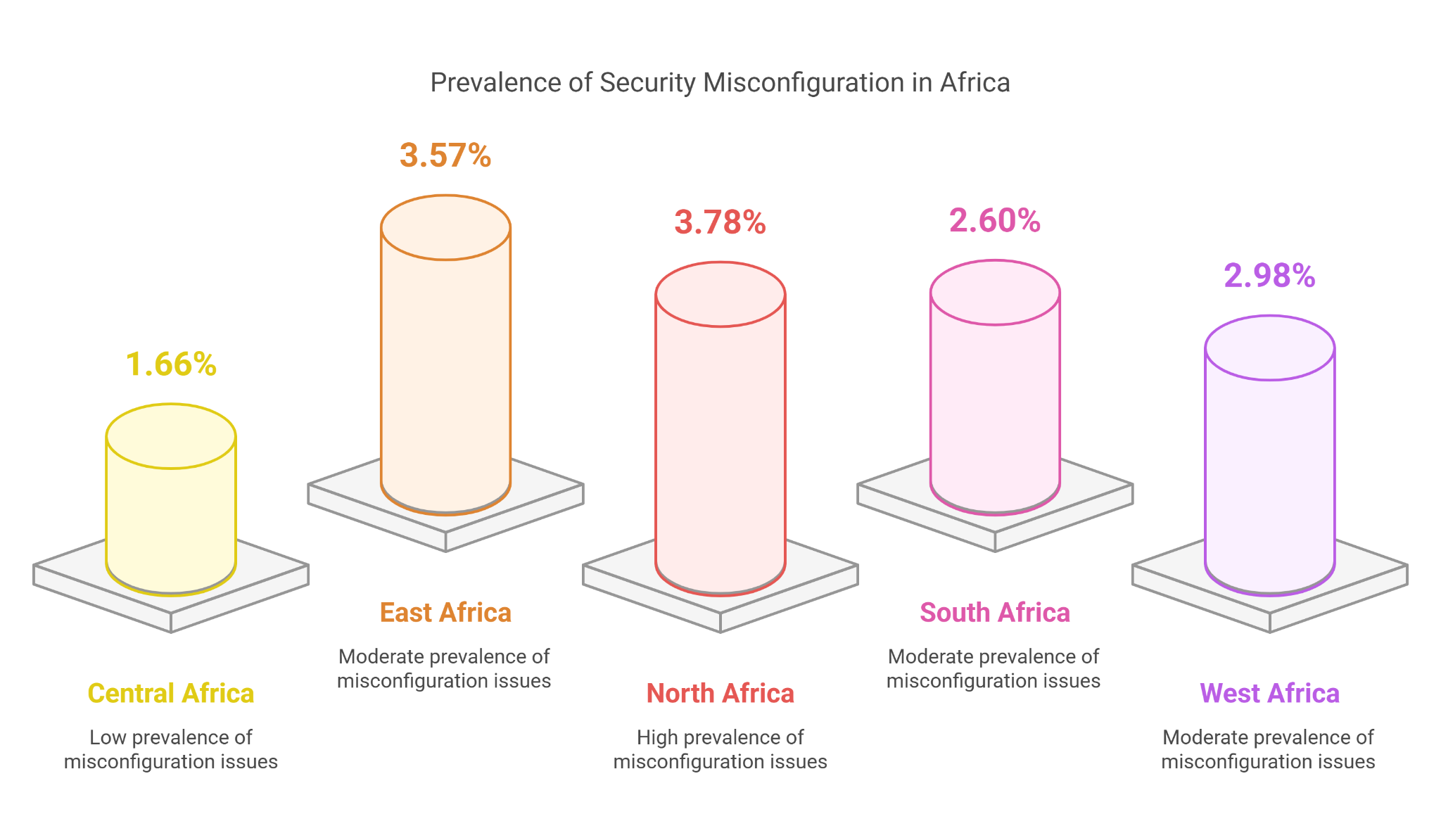
This graph provides an overview of the number of vulnerabilities detected across different African regions. Security misconfiguration is the most prevalent vulnerability, as it is common for developers to leave HTTP methods enabled after debugging. Additionally, its impact is often perceived as less critical compared to injection attacks or sensitive data exposure, which may lead to it being overlooked.

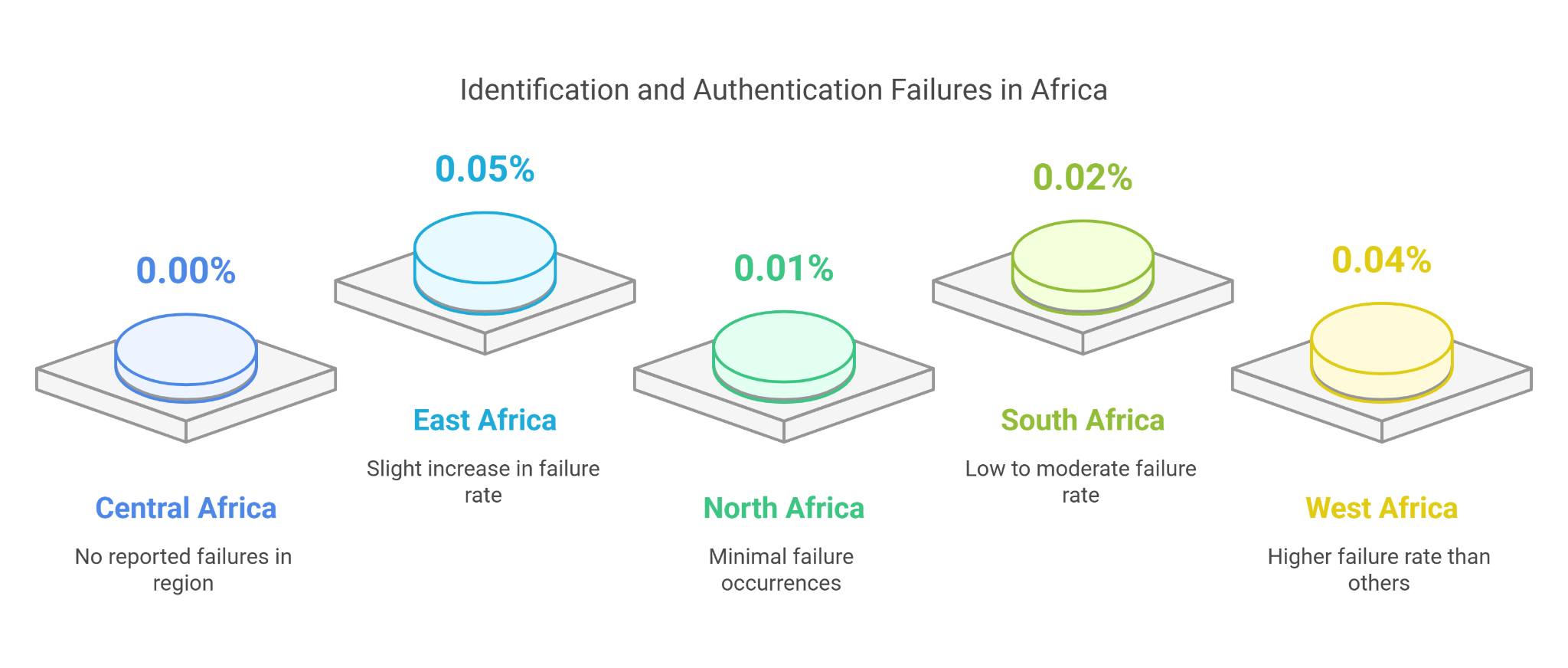
Eastern Africa has the highest number of detected vulnerabilities, primarily due to the larger number of unique URLs tested compared to other regions. Although injection vulnerabilities are not the most frequent, their exploitation is significantly more damaging. However, modern web application frameworks have built-in protections against injection attacks by default, reducing their occurrence.

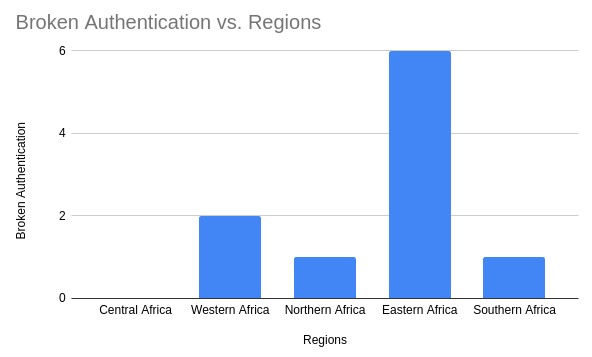
Sensitive data exposure is also widespread, often due to poor file storage practices. Vulnerabilities stem from exposed log files, archive files, and database backups. Developers may also forget to remove sensitive directories, such as .git repositories in the web root, leaving websites vulnerable to data leaks.

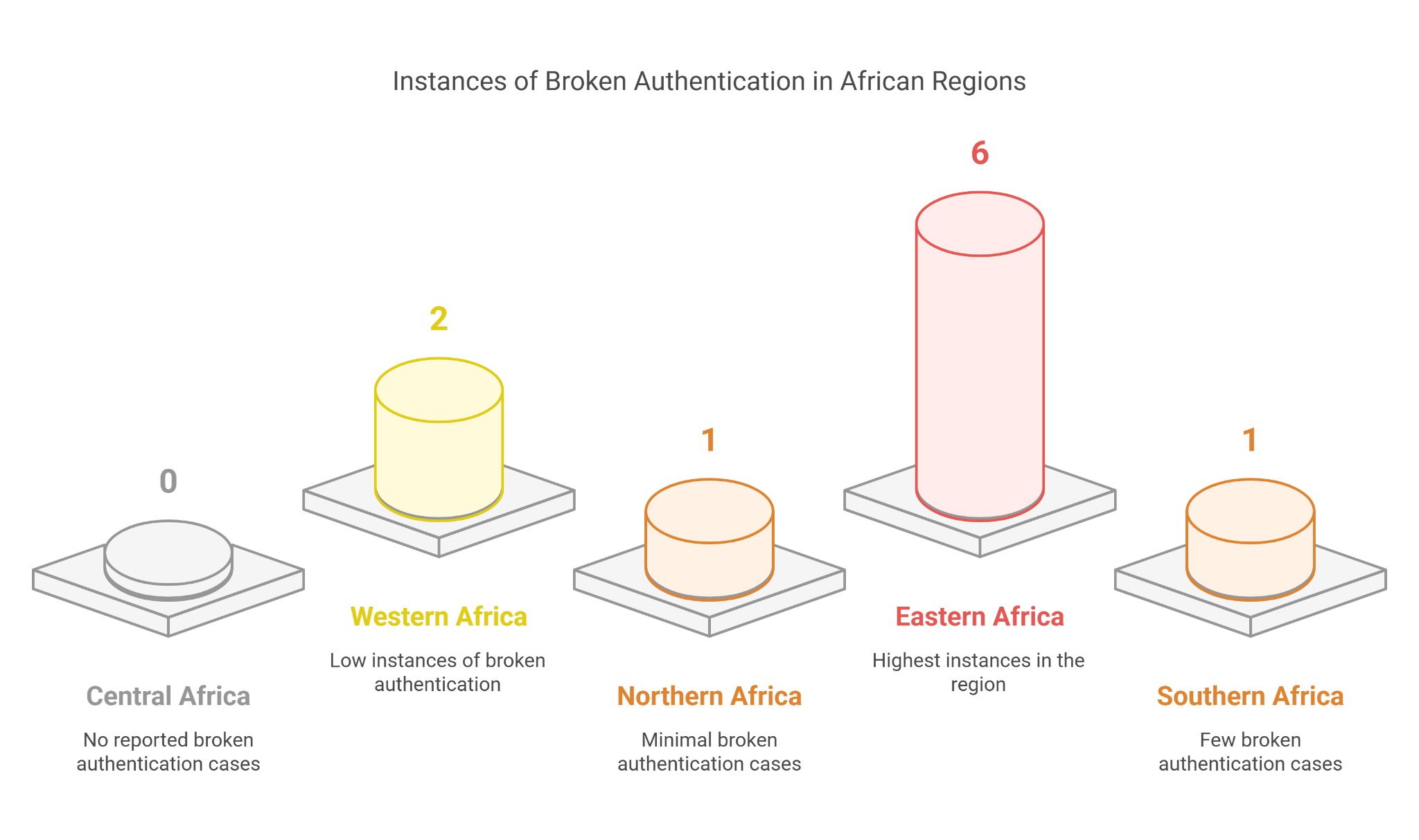


The following graph illustrates the presence of broken authentication vulnerabilities due to weak WordPress passwords, with only 10 vulnerabilities identified.







****

*Figure 12: Diagram instances of Broken Authentication*

# 

# 

# 

# 

# Recommendations and Best Practices

To enhance the security of e-government services, governments should implement multi-factor authentication, enforce strong passwords, and conduct regular vulnerability assessments and penetration tests. Compliance with national and international cybersecurity standards must be ensured through regular checks. Strong encryption is essential to protect sensitive data from unauthorized access. Security updates and patches should be applied promptly, and misconfigurations should be addressed by disabling unnecessary HTTP methods and performing regular audits. Cybersecurity awareness campaigns are also vital, as human error remains a major risk—particularly with ongoing phishing and BEC attacks in Africa. These measures will make government services more secure and resilient to cyber threats.



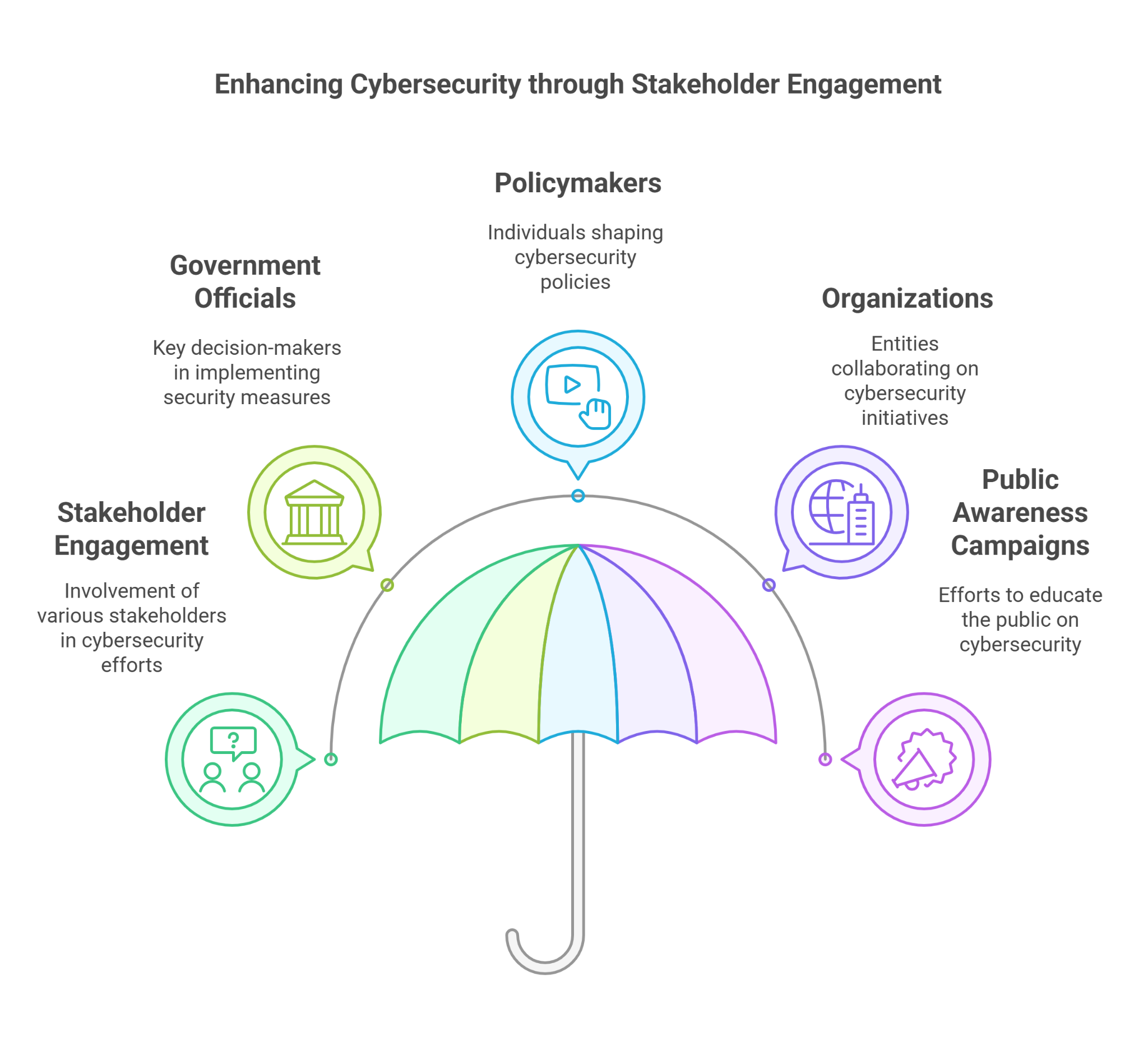
# 

# 

*Figure 13: Diagram summarising the recommendations and best practices to enhance e-government services in Africa*

# Stakeholder Engagement

The findings and recommendations highlighted were communicated to some stakeholders, the government to be more precise, and some vulnerabilities were fixed. The findings reports were sent to 4 countries, and we intend to engage more stakeholders and share reports with them with the recommendations. The findings and recommendations highlighted will be effectively communicated to key stakeholders, including government officials, policymakers, and organizations, to ensure the proper implementation of adequate security measures. These measures align with leveraging expertise for sustainable cybersecurity resilience. In addition, key stakeholders need to engage in public awareness campaigns highlighting the importance of cybersecurity in government e-services, such as educating users on security best practices when interacting with government e-services.



*Figure 14: Diagram depicting the overall stakeholder engagement strategy*

# Forward Look

The vulnerability assessment conducted on the government's e-services has revealed several critical security issues that must be addressed to protect sensitive data and maintain the integrity of these services. The identified vulnerabilities, including Cross-Site Scripting (XSS), insecure HTTP methods, and sensitive content exposure, pose significant risks to users' security and privacy and could potentially lead to severe consequences if exploited by malicious actors. Organizations should remediate these vulnerabilities and implement best security practices.

# 

# 

# 

# References

[1]. <https://carnegieendowment.org/research/2023/10/when-the-rubber-meets-the-road-cybersecurity-and-kenyas-digital-superhighway?utm_source=chatgpt.com&lang=en>

[2]. <https://mybroadband.co.za/news/security/518601-south-african-water-and-sewage-control-systems-potentially-hit-in-global-hack.html>

[3]. <https://www.bbc.com/news/world-africa-68366749>

[4] <https://www.gisreportsonline.com/r/africa-cybersecurity/>

[5]<https://www.diplomacy.edu/resource/report-stronger-digital-voices-from-africa/state-of-african-digital-economy/>

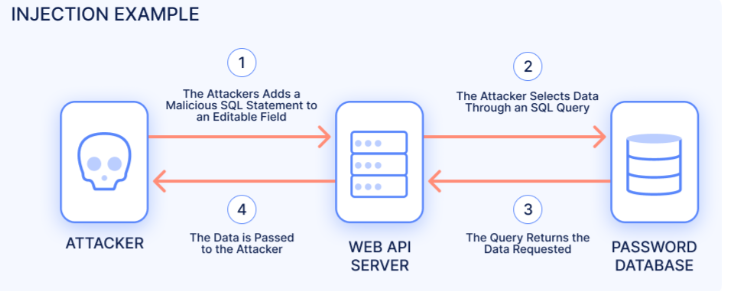
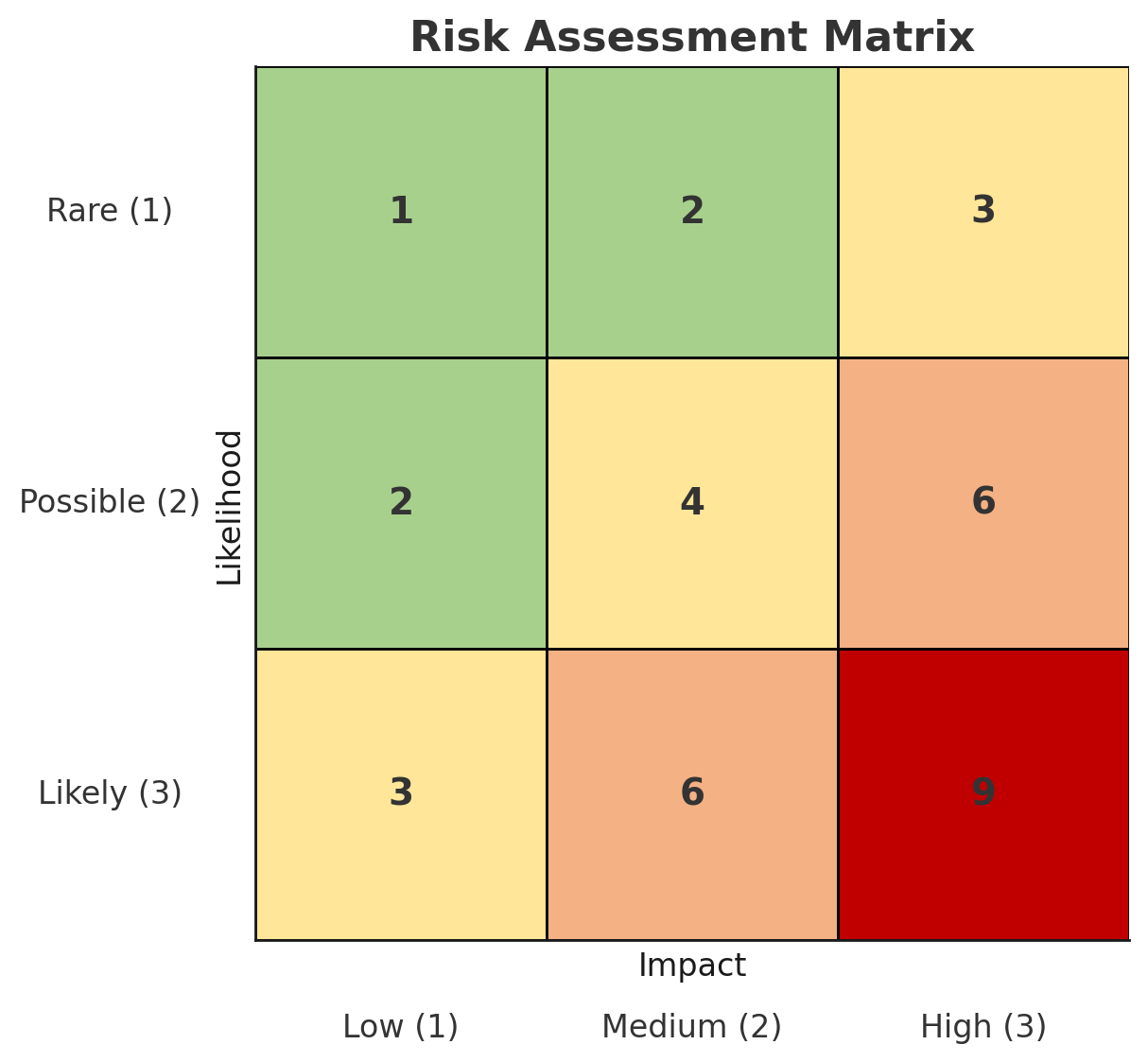
[6] <https://au.int/en/pressreleases/20240522/african-union-strengthens-investigation-capabilities-virtual-assets-and>

[7] <https://www.ionix.io/guides/owasp-top-10/injection-vulnerabilities/>

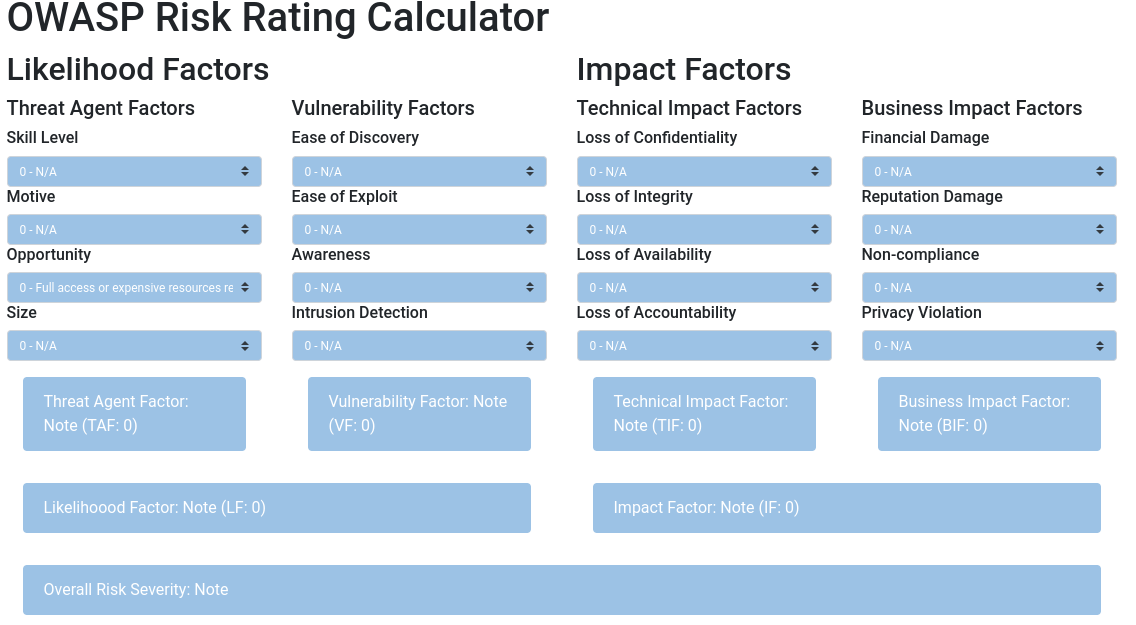
[8] <https://publicadministration.un.org/egovkb/en-us/Data/Region-Information/id/1-Africa>

# 

# Appendices



*Reference:* [*https://www.ionix.io/guides/owasp-top-10/injection-vulnerabilities/*](https://www.ionix.io/guides/owasp-top-10/injection-vulnerabilities/)

**